

图文 071、Java程序员的梦魇：线上系统突然挂掉，可怕的OOM内存溢出！

851 人次阅读 2019-09-09 07:00:00

详情 评论

Java程序员的梦魇：  
线上系统突然挂掉，可怕的OOM内存溢出！



狸猫技术

进店逛

狸猫技术窝专栏上新，基于**真实订单系统**的消息中间件（mq）实战，重磅推荐：



相关频道



从 0 开  
战高手  
已更新1

未来3个月，我的好朋友原子弹大侠将带你一起，全程实战，360度死磕MQ

(点击下方蓝字进行试听)

[从 0 开始带你成为消息中间件实战高手](#)

**重要说明：**

**如何提问：**每篇文章都有评论区，大家可以尽情在评论区留言提问，我都会逐一答疑

(ps：评论区还精选了一些小伙伴对**专栏每日思考题**的作答，有的答案真的非常好！大家可以通过看别人的思路，启发一下自己，从而加深理解)

**如何加群：**购买了狸猫技术窝专栏的小伙伴都可以加入**狸猫技术交流群**。

(群里有不少**一二线互联网大厂**的助教，大家可以一起讨论交流各种技术)

具体**加群方式**请参见文末。

(注：以前通过其他专栏加过群的同学就不要重复加了)

---

## 1、Java程序员的梦魇：线上系统突然挂掉

大家作为一个Java程序员，平时开发系统，测试系统，上线部署系统，为了公司拼命的加班，任劳任怨的干活，然后。。。平时最害怕的是个什么事情？

想必不用我说，大家自己知道了，就是出事故！

很多大公司管事故叫做Case，如果系统一旦出一个事故，比如线上核心系统突然宕机不可用，然后导致几个小时内用户无法下订单，进而导致公司损失几百万，甚至几千万。

或者公司的某个单点登录系统突然不可用，所有用户无法登录APP，也导致无法下单。

或者公司的缓存集群突然全面故障，然后导致公司的全部系统一起瘫痪。

或者因为某个明星突然出轨，结果导致流量集中访问某台服务器，直接把数据库搞挂了。

凡此种种，都是重大的Case。一旦有Case，就会有程序员被拉出去祭天，这是网上常见的一个段子，是不是？

其实祭天不至于那么夸张，但是一旦出了事故，总有人得去承担这个责任，去分析这个事故为什么会发生，谁的责任，后续如何改进

所以，自己负责的线上系统，或者负责维护的缓存集群，或者负责维护的数据库集群，突然莫名其妙挂掉，不可用，导致公司核心业务流程彻底中断，这个就是程序员平时最害怕的事情。

## 2、Java程序员平时最常遇到的故障：系统OOM

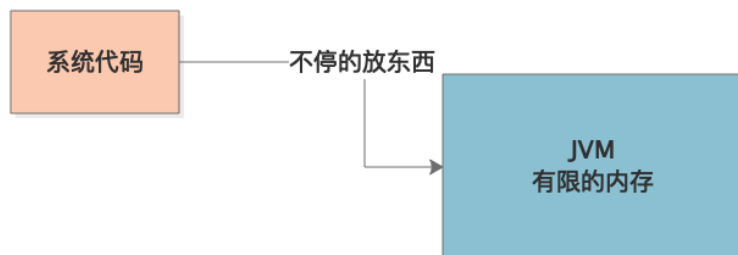
那么作为咱们Java程序员而言，先不考虑自己系统外部依赖的缓存、消息队列、数据库等等东西挂掉，就我们自己系统本身而言，最常见的挂掉的原因是什么？

其实就是系统OOM，也就是所谓的内存溢出！

大家之前跟着我们的专栏学习，已经对JVM的运行原理都比较了解了，那么现在我们就来想一想，所谓的JVM OOM内存溢出到底是什么？

其实说白了，也非常非常的简单，一句话形容，你的JVM内存就这么点，结果你拼命的往里面塞东西，结果内存塞不下了，不就直接溢出了吗？

看看下面的这个图



至于到底JVM是如何放不下对象导致内存溢出的，这个大家不要着急，情况有很多种，我们后面会一步一图一点点给大家分析的

此外我们还会结合一些案例用代码给大家演示出来发生内存溢出的一些场景，最后还会告诉大家平时对线上系统的内存溢出一般怎么来处理 and 解决。

一旦你的系统代码不停的往JVM内存里塞入大量的东西，JVM实在是放不下之后，JVM就会告诉你，OutOfMemory，内存溢出了，我实在放不下那么多东西了，我就直接瘫痪不能工作了。

通常而言，内存溢出这个问题可能对你的系统是毁灭性的打击，他代表你的JVM内存不足以支撑你的代码的运行

所以一旦发生这个情况，就会导致你的系统直接停止运转，甚至会导致你的JVM进程直接崩溃掉，进程都没了！

这个时候对于线上看起来的场景就是，用户突然发现很奇怪，为什么点击APP、点击网页，都没反应了呢？

然后大量的投诉和反馈给到客服，客服直接转移投诉给到运营，运营会直接反馈给技术人员。

这个时候技术人员往往得知这个消息会直接目瞪口呆，最害怕的事情发生了，自己负责的线上系统居然挂掉了，今年的年终奖。。。也许是泡汤了，弄不好还得提前出去找工作去了。。。

### 3、很多工程师都不知道如何处理OOM

最可怕的并不是线上系统发生了OOM，最可怕的是很多工程师压根儿就没有这种处理线上系统故障的经验

当发生OOM之后，根本不知道系统到底为什么会突然OOM？系统代码到底产生了多少对象？为什么会产生这么多对象？JVM为什么会放不下这么多对象？到底怎么去排查这个问题？又如何解决呢？

不知道，全都不知道！

因此我们的专栏接下来的几周内容，将全面围绕JVM OOM的问题展开。

我们先用一周时间带着大家一步一图分析各种可能发生OOM的情况，接着用一周的时间带着大家从一些案例入手来通过模拟代码真实感受一下OOM的发生

接着用一周的时间带着大家去学习如何监控、定位、排查、分析和解决JVM OOM的问题，最后用两周的时间带着大家去用各种真实的生产案例去体验不同场景下的OOM问题。

相信经过专栏最后几周的学习之后，每个人都能游刃有余的处理线上系统的JVM OOM问题的，最终成为JVM的实战高手。

End

专栏版权归公众号**狸猫技术窝**所有

未经许可不得传播，如有侵权将追究法律责任

#### 如何加群？

添加微信号：Lvqu0715\_（微信名：绿小九），狸猫技术窝管理员

发送 Jvm专栏的购买截图

由于是人工操作，发送截图后请耐心等待被拉群

**最后再次提醒：**通过其他专栏加过群的同学，就不要重复加了

**狸猫技术窝其他精品专栏推荐：**

[21天互联网java进阶面试训练营（分布式篇）](#)